For this homework assignment, you will create functions. Each function requires its own Matlab .m file. That means you will be creating multiple files. Please, when you turn in your homework assignment, turn in a single .ZIP file on Blackboard.

**Q1.** Create your own myhistogram() function that replicates the basic functionality of the built-in histogram() function. I want you to tally the bin counts for the histogram using your own code, without using any built-in functions.

Your function should replicate **two** of the calling conventions for the built-in Matlab histogram() command:

```matlab
histogram(data, x);
```

*data* is a one-dimensional array of data values

```matlab
x is a one-dimensional array of the edges each histogram bin
```

```matlab
histogram(data, N);
```

*data* is a one-dimensional array of data values

```matlab
N is the number of equal-sized bins
```

If specifying a one-dimensional array of edges, if you want N bins, you would need to specify N+1 edges; play with the built-in histogram() function to see what it does. To keep things simple, for your code you should check that the edges specify equal-sized bins, giving an error if they do not; note that the built-in histogram() function allows for edges that specify non-equal-sized bins, but you do not need to implement that added complexity.

You can simply assume that if the 2\text{nd} parameter is an array that you are using the first calling convention and if the 2\text{nd} parameter is an integer that you are using the second calling convention.

Your function should create a bar chart of the function like the built-in histogram() function.

To test your myhistogram() function, generate 1000 random numbers drawn from a normal distribution with mean 0 and standard deviation 1. Compare with the built-in histogram() function.

Make sure your code is robust enough to deal gracefully with various combinations of *data* and *x* or *N* that might be passed to your function.
Q2. For the first homework assignment you wrote code that calculated the value of a psychometric function $\psi(x)$ at some level of $x$.

$$\psi(x) = \gamma + (1 - \gamma - \lambda) F(x)$$

$$F(x) = 1 - \exp \left[ -\left( \frac{x}{\alpha} \right)^\beta \right]$$

Assume that $\psi(x)$ is a measure of probability (or proportion) in the range $[0,1]$. $\gamma$ is chance performance (e.g., $\gamma = .5$), $\lambda$ reflects lapses even under the easiest condition (e.g., $\lambda = .05$), and $\alpha$ and $\beta$ reflect the shape of the psychometric function (e.g., $\alpha = 1$ and $\beta = 2$).

The general function is generic in the sense that $F(x)$ can be a Weibull function, as specified above, or it the following Logistic function:

$$F(x) = \frac{1}{1 + \exp \left( -\frac{x - \alpha}{\beta} \right)}$$

(In fact, there are many possible functional forms of $F(x)$, and different applications use different functions.)

Write a function that returns the value of $\psi(x)$ for a specified value of $x$. Your function should also expect a parameter that is a function $F(x)$ that defines the psychometric function (the function for $F(x)$ must be passed to the function that calculates the full equation for the psychometric function). Your function should work whether $x$ is a particular value or $x$ is a one-dimensional array of values.

You should create functions for both of the different forms of $F(x)$.

Illustrate that your function works by calling it with a range of $x$ values that maps out the full shape of the psychometric function $\psi(x)$ and create a plot of the psychometric function using the `plot` command. Do this for both $F(x)$ functions. Make sure you pick a range of $x$ values that fully sweep out the psychometric function given the parameters that define it.

Use $\gamma = 0.5$, $\lambda = 0.05$, $\alpha = 1$, and $\beta = 2$.

Unexcused late assignments will be penalized 10% for every 24 hours late, starting from the time class ends, for a maximum of two days, after which they will earn a 0.